A Topic Modeling Approach to Ranking

Weicong Ding
Boston University

Prakash Ishwar
Boston University

Venkatesh Saligrama
Boston University

Abstract

We propose a topic modeling approach to the prediction of preferences in pairwise comparisons. We develop a new generative model for pairwise comparisons that accounts for multiple shared latent rankings that are prevalent in a population of users. This new model also captures inconsistent user behavior in a natural way. We show how the estimation of latent rankings in the new generative model can be formally reduced to the estimation of topics in a statistically equivalent topic modeling problem. We leverage recent advances in the topic modeling literature to develop an algorithm that can learn shared latent rankings with provable consistency as well as sample and computational complexity guarantees. We demonstrate that the new approach is empirically competitive with the current state-of-the-art approaches in predicting preferences on some semi-synthetic and real world datasets.

1 Introduction

The recent explosion of web technologies has enabled us to collect an immense amount of partial preferences for large sets of items, e.g., products from Amazon, movies from Netflix, or restaurants from Yelp, from a large and diverse population of users through transactions, clicks, check-ins, etc. (e.g., Lu and Boutilier, 2011; Volkovs and Zemel, 2014; Rajkumar and Agarwal, 2014). The goal of this paper is to develop a new approach to model, learn, and ultimately predict the preference behavior of users in pairwise comparisons which can form a building block for other partial preferences. Predicting preference behavior is important to personal recommendation systems, e-commerce, information retrieval, etc.

We propose a novel topic modeling approach to ranking and introduce a new probabilistic generative model for pairwise comparisons that accounts for a heterogeneous population of inconsistent users. The essence of our approach is to view the outcomes of comparisons generated by each user as a *probabilistic mixture* of a few *latent* global rankings that are *shared* across the user-population. This is especially appealing in the context of emerging web-scale applications where (i) there are multiple factors that influence individual preference behavior, e.g., product preferences are influenced by price, brand, etc., (ii) each individual is influenced by multiple latent factors to different extents, (iii) individual preferences for very similar items may be noisy and change with time, and (iv) the number of comparisons available from each user is typically limited. Research on ranking models to-date does not fully capture all these important aspects.

In the literature, we can identify two categories of models. In the first category of models the focus is on learning *one global* ranking that "optimally" agrees with the observations according to some metric (e.g., Gleich and Lim, 2011; Rajkumar and Agarwal, 2014; Volkovs and Zemel, 2014). Loosely speaking, this tacitly presupposes a fairly *homogeneous population* of users having very similar preferences. In the second category of models, there are multiple constituent rankings in the user population, but each user is associated with a *single* ranking scheme sampled from a set of multiple constituent rankings (e.g., Farias et al., 2009; Lu and Boutilier, 2011). Loosely speaking, this tacitly presupposes a *heterogeneous population* of users who are clustered into different types by their preferences and whose preference behavior is influenced by only one factor. In contrast to both these categories, we model each user's pairwise preference behavior as a mixed membership latent variable model. This captures both heterogeneity (via the multiple shared constituent rankings) and inconsistent preference behavior (via the probabilistic mixture). This is a fundamental change of perspective from the traditional clustering-based approach to a decomposition-based one.

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

A second contribution of this paper is the development of a novel algorithmic approach to efficiently and consistently estimate the latent rankings in our proposed model. This is achieved by establishing a formal connection to probabilistic topic modeling where each document in a corpus is viewed as a probabilistic mixture of a few prevailing topics (Blei, 2012). This formal link allows us to leverage algorithms that were recently proposed in the topic modeling literature (Arora et al., 2013; Ding et al., 2013, 2014) for estimating latent shared rankings. Overall, our approach has a running time and a sample complexity bound that are provably *polynomial* in all model parameters. Our approach is asymptotically consistent as the number of users goes to infinity even when the number of comparisons for each user is a small constant.

We also demonstrate competitive empirical performance in collaborative prediction tasks. Through a variety of performance metrics, we demonstrate that our model can effectively capture the variability of real-world user preferences.

2 Related Work

Rank estimation from partial or total rankings has been extensively studied over the last several decades in various settings. A prominent setting is one in which individual user rankings (in a homogeneous population) are modeled as independent drawings from a probability distribution which is centered around a single ground-truth global ranking. Efficient algorithms have been developed to estimate the global ranking under a variety of probability models (Qin et al., 2010; Gleich and Lim, 2011; Negahban et al., 2012; Osting et al., 2013; Volkovs and Zemel, 2014). Chief among them are the Mallows model (Mallows, 1957), the Plackett-Luce (PL) model (Plackett, 1975), and the Bradly-Terry-Luce (BTL) model (Rajkumar and Agarwal, 2014).

To account for the heterogeneity in the user population, (Jagabathula and Shah, 2008; Farias et al., 2009) considered models with multiple prevalent rankings and proposed consistent combinatorial algorithms for estimating the rankings. The mixture of Mallows model recently studied in (Lu and Boutilier, 2011; Awasthi et al., 2014) considers multiple constituent rankings as the "centers" for the Mallows components, as do the "mixture of PL" and the "mixture of BTL" models (Azari Soufiani et al., 2013; Oh and Shah, 2014). In all these settings, however, each user is associated with only one ranking sampled from the mixture model. They capture the cases where the population can be clustered into a few types in terms of their preference behavior.

The setup of our model, although being fundamentally different in modeling perspective, is most closely related to the seminal work in Jagabathula and Shah (2008); Farias et al. (2009) (denoted by FJS) (see Table 1 and appendix). As it turns out, our proposed model subsumes those proposed in FJS as special cases. On the other hand, while the algorithm in FJS can be applied to our more general setting, our algorithm has provably better computational efficiency, polynomial sample complexity, and superior empirical performance.

Relation to topic modeling: Our ranking model shares the same motivation as topic models. Topic modeling has been extensively studied over the last decade and has yielded a number of powerful approaches (e.g., Blei, 2012). While the dominant trend is to fit a MAP/ML estimate using approximation heuristics such as variational Bayes or MCMC, recent work has demonstrated that the topic discovery problem can lend itself to provably efficient solutions with additional structural conditions (Arora et al., 2013; Ding et al., 2014). This forms the basis of our technical approach.

Relation to rating based methods: There is also a considerable body of work on modeling numerical ratings (e.g., Ricci et al., 2011) from which ranking preferences can be derived. An emerging trend explores the idea of combining a topic model for *text reviews* simultaneously with a rating-based model for "star ratings" (Wang and Blei, 2011). These approaches are, however, outside the scope of this paper.

The rest of the paper is organized as follows. We formally introduce the new generative model in Sec. 3. We then present the key geometrical perspective underlying the proposed approach in Sec. 4. We summarize the main steps of our algorithm and the overall computational and statistical efficiency in Sec. 5. We demonstrate competitive performance on semi-synthetic and real-world datasets in Sec. 6.

3 A new generative model

To formalize our proposed model, let $\mathcal{U} := \{1, \dots, Q\}$ be a universe of Q items. Let the K latent rankings over Q items that are shared across a population of M users be denoted by permutations $\sigma^1, \dots, \sigma^K$. Each user compares $N \geq 2$ pairs of items. The *unordered* item pairs $\{i, j\}$ to be compared are assumed to be drawn independently from some distribution μ with $\mu_{i,j} > 0$ for all i, j pairs. The n-th comparison result of user m is denoted by an *ordered* pair $w_{m,n} = (i, j)$, if user m compares item i and j and prefers i over j. Let a probability vector $\boldsymbol{\theta}_m$ be the user-specific weights over the K latent rankings. The generative model for the comparisons from each user $m = 1, \dots, M$ is,

1. Sample $\boldsymbol{\theta}_m \in \triangle^K$ from a prior distribution $\Pr(\theta)$

Table 1: Comparison to closely related work (Jagabathula and Shah, 2008) (Farias et al., 2009) (FJS)

Method	**Assumptions on $\boldsymbol{\sigma}$**	**Statistics used**	**Consistency proved?**	**Computational complexity**	**Sample complexity**
FJS	Separability	1st order	Yes	Exponential in K	Not provided
This paper	Separability	up to 2nd order	Yes	Polynomial	Polynomial

2. For each comparison $n = 1, \ldots, N$:
   (a) Sample a pair of items $\{i, j\}$ from μ
   (b) Sample a ranking token $z_{m,n} \in \{1, \ldots, K\} \sim$ Multinomial$(\boldsymbol{\theta}_m)$
   (c) If $\sigma^{z_{m,n}}(i) < \sigma^{z_{m,n}}(j)$, then $w_{m,n} = (i, j)$, otherwise $w_{m,n} = (j, i)$ [1]

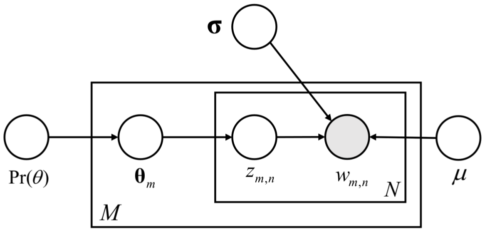

Figure 1: Graphical model representation of the generative model. The boxes represent replicates. The outer plate represents users, and the inner plate represents ranking tokens and comparisons of each user.

Figure 1 is a standard graphical model representation of the proposed generative process. Each user is characterized by $\boldsymbol{\theta}_m$, the user-specific weights over the K shared rankings. For convenience, we represent $\sigma^1, \ldots, \sigma^K$ by a $W \times K$ nonnegative ranking matrix $\boldsymbol{\sigma}$ whose $W = Q(Q-1)$ rows are indexed by all the ordered pairs (i, j). We set $\sigma_{(i,j),k} = \mathbb{I}(\sigma^k(i) < \sigma^k(j))$, so that the k-th column of $\boldsymbol{\sigma}$ is an equivalent representation of the ranking σ^k. We then denote by $\boldsymbol{\theta}$ the $K \times M$ dimensional weight matrix whose columns are the user-specific mixing weights $\boldsymbol{\theta}_m$'s. Finally, let $\mathbf{X}$ be the $W \times M$ empirical comparisons-by-user matrix where $X_{(i,j),m}$ denotes the number of times that user m compares pair $\{i, j\}$ and prefers item i over j. The principal algorithmic problem is to estimate the ranking matrix $\boldsymbol{\sigma}$ given $\mathbf{X}$ and K.

If we denote by $\mathbf{P}$ a $W \times W$ diagonal matrix with the (i, j)-th diagonal component $P_{(i,j),(i,j)} = \mu_{i,j}$, and set $\mathbf{B} = \mathbf{P}\boldsymbol{\sigma}$, then the generative model induces the following probabilities on comparisons $w_{m,n}$:

$$
\begin{aligned}
p(w_{m,n} = (i,j)|\boldsymbol{\theta}_m, \boldsymbol{B}) &= \mu_{i,j} \sum_{k=1}^{K} \sigma_{(i,j),k}\theta_{k,m} \\
&= \sum_{k=1}^{K} B_{(i,j),k}\theta_{k,m} \qquad (1)
\end{aligned}
$$

Similarly, if we consider a probabilistic topic model on a set of M documents, each composed of N words drawn from a vocabulary of size W, with a $W \times K$ *topic* matrix $\boldsymbol{\beta}$ and document-specific mixing weights $\boldsymbol{\theta}_m^{\mathrm{TM}}$ sampled from a topic prior $\Pr^{\mathrm{TM}}(\theta)$ (e.g. Blei, 2012), then, the distribution induced on the observation $w_{m,n}^{\mathrm{TM}}$, i.e., the n-th word in document m, has the same form as in (1):

$$
p(w_{m,n}^{\mathrm{TM}} = i|\boldsymbol{\theta}_m^{\mathrm{TM}}, \boldsymbol{\beta}) = \sum_{k=1}^{K} \beta_{i,k}\theta_{k,m}^{\mathrm{TM}} \qquad (2)
$$

where $i = 1, \ldots, W$ is any distinct word in the vocabulary. Noting that $\mathbf{B}$ is column-stochastic, we have,

Lemma 1. *The proposed generative model is statistically equivalent to a standard topic model whose topic matrix $\boldsymbol{\beta}$ is set to be $\mathbf{B}$ and the topic prior to be $\Pr(\theta)$.*

Proof. Note that since $\mathbf{B}$ is column stochastic, it is a valid topic matrix. We need to show that the distribution on the comparisons $\mathbf{w} = \{w_{m,n}\}$ and on the words in topic model $\mathbf{w}^{\mathrm{TM}} = \{w_{m,n}^{\mathrm{TM}}\}$ are the same. From (1) (2),

$$
\begin{aligned}
p(\mathbf{w}|\mathbf{B}) &= \prod_{m=1}^{M} \int p(w_{m,1}, \ldots, w_{m,N}|\boldsymbol{\theta}_m, \mathbf{B}) \Pr(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m \\
&= \prod_{m=1}^{M} \int \left(\prod_{n=1}^{N} \sum_{k=1}^{K} B_{w_{m,n},k}\theta_{k,m} \right) \Pr(\boldsymbol{\theta}_m) d\boldsymbol{\theta}_m \\
&= p(\mathbf{w}^{\mathrm{TM}}|\boldsymbol{\beta}).
\end{aligned}
$$

□

Note that $\boldsymbol{B} = \mathbf{P}\boldsymbol{\sigma}$, $\mu_{i,j} = \mu_{j,i}$, and $\sigma_{(i,j),k} + \sigma_{(j,i),k} = 1$. Hence $\boldsymbol{\sigma}$ can be inferred directly from $\mathbf{B}$:

$$
\sigma_{(i,j),k} = \frac{\sigma_{(i,j),k}\mu_{i,j}}{(\sigma_{(i,j),k} + \sigma_{(j,i),k})\mu_{i,j}} = \frac{B_{(i,j),k}}{B_{(i,j),k} + B_{(j,i),k}} \qquad (3)
$$

Thus, the problem of estimating the ranking matrix $\boldsymbol{\sigma}$ can be solved by any approach that can learn the topic matrix $\boldsymbol{\beta}$. Our approach is to leverage recent works in topic modeling (Arora et al., 2012, 2013; Ding et al., 2013, 2014) that come with consistency and statistical and computational efficiency guarantees by exploiting the second-order moments of the columns of $\mathbf{X}$, i.e., a co-occurrence matrix of pairwise comparisons. We can establish parallel results for ranking model via the equivalency result of Lemma 1. In particular, by combining Lemma 1 with results in (Ding et al., 2013, Lemma 1 in Appendix), the following result can be immediately established:

[1] $\sigma^k(i)$ is the position of item i in the ranking σ_k and item i is preferred over j if $\sigma^k(i) < \sigma^k(j)$.

Lemma 2. *If $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{X}}'$ are obtained from $\mathbf{X}$ by first splitting each user's comparisons into two independent copies and then re-scaling the rows to make them row-stochastic, then*

$$M\widetilde{\mathbf{X}}'\widetilde{\mathbf{X}}^\top \xrightarrow[\textit{almost surely}]{M\to\infty} \bar{\boldsymbol{B}}\bar{\mathbf{R}}\bar{\boldsymbol{B}}^\top =: \mathbf{E}, \qquad (4)$$

where $\bar{\boldsymbol{B}} = \operatorname{diag}^{-1}(\boldsymbol{B}\mathbf{a})\boldsymbol{B}\operatorname{diag}(\mathbf{a})$, $\boldsymbol{B} = \mathbf{P}\boldsymbol{\sigma}$, $\bar{\mathbf{R}} = \operatorname{diag}^{-1}(\mathbf{a})\mathbf{R}\operatorname{diag}^{-1}(\mathbf{a})$, and $\mathbf{a}$ and $\mathbf{R}$ are, respectively, the $K \times 1$ expectation and $K \times K$ correlation matrix of the weight vector $\boldsymbol{\theta}_m$.

4 A Geometric Perspective

$\sigma^1 : 1 > 2 > 3$
$\sigma^2 : 2 > 3 > 1$
$\sigma^3 : 3 > 1 > 2$

$$\begin{array}{c} \\ (1,2) \\ (1,3) \\ (2,1) \\ (2,3) \\ (3,1) \\ (3,2) \end{array} \begin{array}{c} \begin{array}{ccc} \text{Rank 1} & \text{Rank 2} & \text{Rank 3} \end{array} \\ \left[\begin{array}{ccc} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}\right] \\ \boldsymbol{\sigma} \end{array}$$

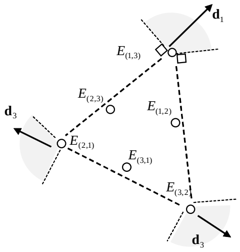

Figure 2: A separable ranking matrix $\boldsymbol{\sigma}$ with $K = 3$ rankings over $Q = 3$ items, and the underlying geometry of the row vectors of $\mathbf{E}$. $(1,3), (2,1), (3,2)$ are novel pairs. Shaded regions depict the solid angles of the extreme points.

The key insight of our approach is an intriguing geometric property of the normalized second-order moment matrix $\mathbf{E}$ (defined in Lemma 2) illustrated in Fig. 2. This arises from the so-called *separability condition* on the ranking matrix $\boldsymbol{\sigma}$,

Definition 1. *A ranking matrix $\boldsymbol{\sigma}$ is separable if for each ranking k, there is at least one ordered pair (i, j), such that $\sigma_{(i,j),k} > 0$ and $\sigma_{(i,j),l} = 0$, $\forall\ l \neq k$.*

In other words, for each ranking, there exists at least one "novel" pair of items $\{i, j\}$ such that i is uniquely preferred over j in that ranking while j is ranked higher than i in all the other rankings. Figure 2 shows an example of a separable ranking matrix in which the ordered pair $(1, 3)$ is novel to ranking $\boldsymbol{\sigma}^1$, the pair $(2, 1)$ to $\boldsymbol{\sigma}^2$, and the pair $(3, 2)$ to $\boldsymbol{\sigma}^3$.

The separability condition has been identified as a good approximation for real-world datasets in nonnegative matrix factorization (Donoho and Stodden, 2004) and topic modeling (Arora et al., 2013; Ding et al., 2014), etc. In the context of ranking, this condition has appeared, albeit implicitly in a different form, in the seminal works of (Jagabathula and Shah, 2008; Farias et al., 2009). Moreover, as shown in (Farias et al., 2009), the separability condition is satisfied with high probability when the $K \ll Q$ underlying rankings are sampled uniformly from the set of all $Q!$ permutations. In our experiments we have observed that the ranking matrix induced by the rating matrix estimated by matrix factorization is often separable (Sec. 6.2).

If $\boldsymbol{\sigma}$ is separable then the novel pairs correspond to extreme points of the convex hull formed by all the row vectors of $\mathbf{E}$ (Fig. 2). Thus, the novel pairs can be efficiently identified through an extreme point finding algorithm. Once all the novel pairs are identified, the ranking matrix can be estimated using a constrained linear regression (Arora et al., 2013; Ding et al., 2014). To exclude redundant rankings and ensure unique identifiability, we assume $\mathbf{R}$ has full rank.

We leverage the normalized **Solid Angle** subtended by extreme points to detect the novel pairs as proposed in (Ding et al., 2014, Definition 1). The solid angles are indicated by the shaded regions in Fig. 2. From a statistical viewpoint, it can be defined as the probability that a row vector $\mathbf{E}_{(i,j)}$ has the maximum projection value along an isotropically distributed random direction $\mathbf{d}$:

$$\begin{aligned} q_{(i,j)} \triangleq p\{\forall (s,t) : \mathbf{E}_{(i,j)} \neq \mathbf{E}_{(s,t)}, \\ \langle \mathbf{E}_{(i,j)}, \mathbf{d}\rangle > \langle \mathbf{E}_{(s,t)}, \mathbf{d}\rangle\} \qquad (5) \end{aligned}$$

These can be efficiently approximated using a few iid isotropic $\mathbf{d}$'s. By following the approach in (Ding et al., 2014, Lemma 2) for topic modeling, one can prove the following result which shows that the solid angles can be used to detect novel pairs:

Lemma 3. *Suppose $\boldsymbol{\sigma}$ is separable and $\mathbf{R}$ is full rank, then, $q_{(i,j)} > 0$ if and only if (i, j) is a novel pair.*

This motivates the following solution approach: (1) Estimate the solid angles $q_{(i,j)}$, (2) Select K distinct pairs with largest $q_{(i,j)}$'s, and (3) Estimate the ranking matrix $\boldsymbol{\sigma}$ using constrained linear regression.

Given the estimated ranking matrix $\boldsymbol{\sigma}$ (and $\mathbf{B}$), we follow the typical steps in topic modeling (Blei, 2012) to fit the ranking prior, infer user-specific preferences $\boldsymbol{\theta}_m$, and predict new comparisons (see Sec. 6).

5 Algorithm and Analysis

The main steps of our approach are outlined in Algorithm 1 and expanded in detail in Algorithms 2, 3 and 4. Algorithm 2 detects all the novel pairs for the K distinct rankings. Once the novel pairs are identified, Algorithm 3 estimates matrix $\mathbf{B}$ using constrained linear regression followed by row and then column scaling.

Algorithm 4 further processes $\widehat{\mathbf{B}}$ to obtain an estimate of the ranking matrix $\boldsymbol{\sigma}$. Step 1 is based on Eq. (3) and step 2 further rounds each element to 0 or 1. Algorithm 4 guarantees that $\widehat{\boldsymbol{\sigma}}$ is binary and satisfies the condition: $\widehat{\sigma}_{(i,j),k} + \widehat{\sigma}_{(j,i),k} = 1$ for all $i \neq j$ and all k.

Algorithm 1 Ranking Recovery (Main Steps)

```
Input: Pairwise comparisons $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{X}}'(W \times M)$; Number
    of rankings $K$; Number of projections $P$; Tolerance
    parameters $\zeta, \epsilon > 0$.
Output: Ranking matrix estimate $\widehat{\boldsymbol{\sigma}}$.
  Novel Pairs $\mathcal{I} \leftarrow$NovelPairDetect($\widetilde{\mathbf{X}}, \widetilde{\mathbf{X}}', K, P, \zeta$)
  $\widehat{\mathbf{B}} \leftarrow$EstimateRankings($\mathcal{I}, \mathbf{X}, \epsilon$)
  $\widehat{\boldsymbol{\sigma}} \leftarrow$PostProcess($\widehat{\mathbf{B}}$)
```

Algorithm 2 NovelPairDetect (via Random Projections)

```
Input: $\widetilde{\mathbf{X}}$, $\widetilde{\mathbf{X}}'$; number of rankings $K$; number of pro-
    jections $P$; tolerance $\zeta$;
Output: $\mathcal{I}$: The set of all novel pairs of $K$ distinct
    rankings.
  $\widehat{\mathbf{E}} \leftarrow M\widetilde{\mathbf{X}}'\widetilde{\mathbf{X}}^\top$
  $\forall(i,j),\ \mathcal{J}_{(i,j)} \leftarrow \{(s,t) : \widehat{E}_{(i,j),(i,j)} - 2\widehat{E}_{(i,j),(s,t)} + \widehat{E}_{(s,t),(s,t)} \geq \zeta/2\}$,
  for $r = 1, \dots, P$ do
    Sample $\mathbf{d}_r \in \mathbb{R}^W$ from an isotropic prior
    $\hat{q}_{(i,j),r} \leftarrow \mathbb{I}\{\forall(s,t) \in \mathcal{J}_{(i,j)},\ \widehat{\mathbf{E}}_{(s,t)}\mathbf{d}_r \leq \widehat{\mathbf{E}}_{(i,j)}\mathbf{d}_r\}$ , $\forall(i,j)$
  end for
  $\hat{q}_{(i,j)} \leftarrow \frac{1}{P}\sum_{r=1}^{P} \hat{q}_{(i,j),r}, \forall(i,j)$
  $k \leftarrow 0, l \leftarrow 1$, and $\mathcal{I} \leftarrow \emptyset$
  while $k \leq K$ do
    $(s,t) \leftarrow$ index of the $l^{\text{th}}$ largest value among $\hat{q}_{(i,j)}$'s
    if $(s,t) \in \bigcap_{(i,j)\in\mathcal{I}} \mathcal{J}_{(i,j)}$ then
      $\mathcal{I} \leftarrow \mathcal{I} \cup \{(s,t)\}, \quad k \leftarrow k+1$
    end if
    $l \leftarrow l+1$
  end while
```

Algorithm 3 Estimate Rankings

```
Input: $\mathcal{I} = \{(i_1,j_1), \dots, (i_K,j_K)\}$ the set of novel
    pairs of $K$ rankings; $\mathbf{X}$, $\mathbf{X}'$; precision $\epsilon$
Output: $\widehat{\boldsymbol{B}}$ as the estimate of $\boldsymbol{B}$.
  $\mathbf{Y} = (\widetilde{\mathbf{X}}_{(i_1,j_1)}^\top, \dots, \widetilde{\mathbf{X}}_{(i_K,j_K)}^\top)^\top$,
  $\mathbf{Y}' = (\widetilde{\mathbf{X}}_{(i_1,j_1)}'^\top, \dots, \widetilde{\mathbf{X}}_{(i_K,j_K)}'^\top)^\top$
  for all $(i,j)$ pairs do
    Solve $\widehat{\boldsymbol{\beta}}_{(i,j)} \leftarrow \arg\min_{\mathbf{b}} M(\widetilde{\mathbf{X}}_{(i,j)} - \mathbf{b}\mathbf{Y})(\widetilde{\mathbf{X}}'_{(i,j)} - \mathbf{b}\mathbf{Y}')^\top$
    Subject to $b_k \geq 0, \sum_{k=1}^{K} b_k = 1$, With precision $\epsilon$
    $\widehat{\boldsymbol{\beta}}_{(i,j)} \leftarrow (\frac{1}{M}\mathbf{X}_{(i,j)}\mathbf{1})\widehat{\boldsymbol{\beta}}_{(i,j)}$
  end for
  $\widehat{\boldsymbol{B}} \leftarrow$column normalize $\widehat{\boldsymbol{\beta}}$
```

Our approach inherits the polynomial computational complexity of the topic modeling algorithm in Ding et al. (2014):

Algorithm 4 Post Processing

```
Input: $\widehat{\mathbf{B}}$ as the estimate of $\mathbf{B}$
Output: $\widehat{\boldsymbol{\sigma}}$ as the estimate of $\boldsymbol{\sigma}$
  $\widehat{\sigma}_{(i,j),k} \leftarrow \frac{\widehat{B}_{(i,j),k}}{\widehat{B}_{(i,j),k} + \widehat{B}_{(j,i),k}}, \forall i,j \in \mathcal{U}, \forall k$
  $\widehat{\sigma}_{(i,j),k} \leftarrow \text{Round}[\widehat{\sigma}_{(i,j),k}], \forall i,j \in \mathcal{U}, \forall k$
```

Theorem 1. *The running time of Algorithm 1 is $\mathcal{O}(MNK + Q^2K^3)$.*

We further derive the sample complexity bounds for our approach which is also polynomial in all model parameters and $\log(1/\delta)$ where δ is the upper bound on error probability. A major technical improvement compared to the results that appear in Ding et al. (2014) is that our analysis holds true for *any isotropic distribution* over the random directions $\mathbf{d}$ in Alg. 2. The previous result in (Ding et al., 2014, Theorem 1, 2) was designed only for specific distributions such as spherical Gaussian. Formally,

Theorem 2. *Let the ranking matrix $\boldsymbol{\sigma}$ be separable and $\mathbf{R}$ have full rank. Then the Algorithm 1 can consistently recover $\boldsymbol{\sigma}$ up to a column permutation as the number of users $M \to \infty$ and number of projections $P \to \infty$. Furthermore, for any isotropically drawn random direction $\mathbf{d}$, $\forall\delta > 0$, if*

$$M \geq \max\left\{40\frac{\log(3W/\delta)}{N\rho^2\eta^4},\ 320\frac{W^{0.5}\log(3W/\delta)}{N\eta^6\lambda_{\min}}\right\}$$

and $P \geq 16\frac{\log(3W/\delta)}{q_\wedge^2}$, then Algorithm 1 fails with probability at most δ. The other model parameters are defined as $\eta = \min_{1\leq w\leq W}[\mathbf{Ba}]_w$, $\rho = \min\{\frac{d}{8}, \frac{\pi d_2 q_\wedge}{4W^{1.5}}\}$, $d_2 \triangleq (1-b)\lambda_{\min}$, $d = (1-b)^2\lambda_{\min}^2/\lambda_{\max}$, $b = \max_{j\in\mathcal{C}_0,k}\bar{B}_{j,k}$ and $\lambda_{\min}$, $\lambda_{\max}$ are the minimum /maximum eigenvalues of $\bar{\mathbf{R}}$. $q_\wedge$ is the minimum solid angle of the extreme points of the convex hull of the rows of $\mathbf{E}$.

Detailed proofs are provided in the supplementary material. We combine the analysis of Alg. 4 and the rescaling steps in Alg. 3 in order to exploit the structural constraints of the ranking model. As a result, we obtain an improved sample complexity bound for M compared to Ding et al. (2014); Arora et al. (2013)

6 Experimental Validation

6.1 Overview of Experiments and Methodology

We conduct experiments first on semi-synthetic dataset in order to validate the performance of our proposed algorithm when the model assumptions are satisfied, and then on real-world datasets in order to demonstrate that the proposed model can indeed effectively capture the variability that one encounters in the real world. We focus on the collaborative filtering applications where population heterogeneity and user inconsistency are the well-known characteristics (e.g., Salakhutdinov and Mnih, 2008a).

We use Movielens, a benchmark movie-rating dataset widely used in the literature.[2] The rating-based data is selected due to its public availability and widespread use, but we convert it to pairwise comparisons data and focus on modeling from a ranking viewpoint. This procedure has been suggested and widely used in the rank-aggregation literature (e.g., Lu and Boutilier, 2011; Volkovs and Zemel, 2014). For the semi-synthetic datasets, we evaluate the **reconstruction error** between the learned rankings $\widehat{\boldsymbol{\sigma}}$ and the ground truth. We adopt the standard *Kendall's tau distance* between two rankings. For the real-world datasets where true parameters are not available, we use the **held-out log-likelihood**, a standard metric in ranking prediction (Lu and Boutilier, 2011) and in topic modeling Wallach et al. (2009).

In addition, we consider the standard task of rating prediction via our proposed ranking model. Our aim here is to illustrate that our model is suitable for real-word data. We do not optimize tuning parameters in order to achieve the best result. We measure the performance by **root-mean-square-error** (RMSE) which is the standard in literature(e.g., Salakhutdinov and Mnih, 2008a; Toscher et al., 2009).

The parameters of our algorithm are the same as in Ding et al. (2014). Specifically, the number of random projections $P = 150 \times K$, the tolerance parameter $\zeta/2$ for Alg. 2 is fixed at 0.01 and the precision parameter $\epsilon = 10^{-4}$ for Alg. 3.

6.2 Semi-synthetic simulation

We first use a *semi-synthetic* dataset to validate the performance of our algorithm. In order to match the dimensionality and other characteristics that are representative of real-world examples, we generate the semi-synthetic pairwise comparisons dataset using a benchmark movie star-ratings dataset, Movielens. The original dataset has approximately 1 million ratings for 3952 movies from $M = 6040$ users. The ratings range from 1 star to 5 stars.

We follow the procedure in (Lu and Boutilier, 2011) and (Volkovs and Zemel, 2014) to generate the semi-synthetic dataset as follows. We consider the $Q = 100$ most frequently rated movies and train a latent factor model on the star-ratings data using a state-of-the-art matrix factorization based algorithm (Salakhutdinov and Mnih, 2008a). This approach is selected for its state-of-the-art performance on many real-world collaborative filtering tasks. This procedure learns a $Q \times K$ movie-factor matrix whose columns are interpreted as scores of the Q movies over the K latent factors(Salakhutdinov and Mnih, 2008a; Volkovs and Zemel, 2014). By sorting the scores of each column of the movie-factor matrix, we obtain K rankings for generating the semi-synthetic dataset. We set $K = 10$ as suggested by Lu and Boutilier (2011) and Salakhutdinov and Mnih (2008a). We note that the resulting ranking matrix $\boldsymbol{\sigma}$ satisfies the separability condition.

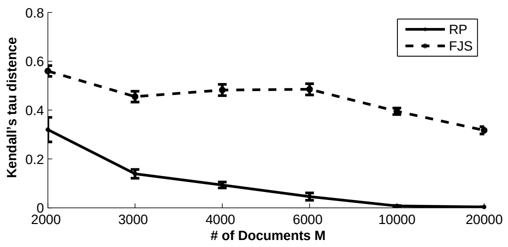

Figure 3: The normalized Kendall's tau distance error of the estimated rankings, as functions of M, estimated by RP and FJS from the semi-synthetic dataset with $Q = 100, N = 300, K = 10$.

The other model parameters are set as follows. $\mu_{i,j} = 1/\binom{Q}{2}$, $\forall i,j \in \mathcal{U}$. The prior distribution for $\boldsymbol{\theta}_m$ is set to be Dirichlet $\Pr(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) = \frac{1}{C}\prod_{k=1}^{K}\theta_k^{\alpha_k - 1}$ as suggested by (Lu and Boutilier, 2011). The parameters α_k's are determined by $\alpha_k = \alpha_0 a_k$, where the concentration parameter $\alpha_0 = 0.1$ and the expectation $\mathbf{a} = [a_1, \ldots, a_K]^\top$ is sampled uniformly from the $K = 10$ dimensional simplex for each random realization. We note that the correlation matrix $\mathbf{R}$ of the Dirichlet distribution has full rank (Arora et al., 2013). We fix $N = 300$ comparisons per user to approximate the observed average pairwise comparisons in the Movielens dataset and vary M.

Since the output of our algorithm is determined only up to a column permutation, we first align the columns of $\boldsymbol{\sigma}$ and $\widehat{\boldsymbol{\sigma}}$ using bipartite matching based on ℓ_1 distance, and then measure the performance by the ℓ_1 distance between the ground truth rankings $\boldsymbol{\sigma}$ and the estimate $\widehat{\boldsymbol{\sigma}}$. Due to the way $\boldsymbol{\sigma}$ is defined, this is equivalent to the widely-used *Kendall's tau distance* between two rankings which is proportional to the number of pairs in which two ranking schemes differ. We further normalize the ℓ_1 error by $W = Q \times (Q - 1)$ so that the error measure for each column is a number between $[0, 1]$.

[2] Another large benchmark, Netflix dataset is not available due to privacy issues. Movielens is currently available at `http://grouplens.org/datasets/movielens/`

We compare our proposed algorithm (denoted by RP) against the algorithm proposed in (Jagabathula and Shah, 2008; Farias et al., 2009) (denoted by FJS) for estimating the ranking matrix. To the best of our knowledge, this is the most recent algorithm with consistency guarantees for $K > 1$.[3] We compared how the estimation error varies with the number of users M, and the results are depicted in Fig. 3. For each setting, we average over 10 Monte Carlo runs. Evidently, our algorithm shows superior performance over FJS. More specifically, since our ground truth ranking matrix is separable, as M increases, the estimation error of RP converges to zero, and the convergence is much faster than FJS. We note that only for $M \geq 100,000$ does the error of the FJS algorithm eventually start approaching 0.

6.3 Movielens - Comparison prediction

We apply the proposed algorithm (RP) to the real-world Movielens dataset introduced in Sec. 6.2 and consider the task of predicting pairwise comparisons. We consider two settings: (1) new comparison prediction, and (2) new user prediction. We train and evaluate our model using the comparisons obtained from the star-ratings of the Movielens dataset. This procedure of generating comparisons from star-ratings is motivated by (Lu and Boutilier, 2011; Volkovs and Zemel, 2014). We focus on the $Q = 100$ most frequently rated movies and obtain a subset of 183,000 star-ratings from $M = 5940$ users. The pairwise comparisons are generated from the star ratings following (Lu and Boutilier, 2011; Volkovs and Zemel, 2014): for each user m, we **select** pairs of movies i, j that user m rated, and **compare** the stars of the two movies to generate comparisons.

To **select** pairs of items to compare, we consider: (a) (Full) all pairs of movies that a user has rated, or (b) (Partial) randomly select $5N_{star,m}$ pairs where $N_{start,m}$ is the number of movies user m has rated.

To **compare** a pair of movies i, j rated by a user, $w_{m,n} = (i, j)$ if the star rating of i is higher than j. For ties, we consider: (i)(Both) generate $w_{m,1} = (i, j)$ and $w_{m,2} = (j, i)$, (ii) (Ignore) do nothing, and (iii) (Random) select one of $w_{m,1}, w_{m,2}$ with equal probability.

New comparison prediction: In this setting, for each user, a subset of her ratings are used to generate the training comparisons while the remaining are for testing comparisons. We follow the training/testing split as in (Salakhutdinov and Mnih, 2008a).[4] We convert both the training ratings and testing ratings into training comparisons and testing comparisons independently.

We evaluate the performance by the predictive log-likelihood of the testing data, i.e., $\Pr(\mathbf{w}_{test}|\mathbf{w}_{train}, \widehat{\boldsymbol{\sigma}})$. Given the estimate $\widehat{\boldsymbol{\sigma}}$, we follow (Arora et al., 2013; Ding et al., 2014) to fit a Dirichlet prior model. We then calculate the prediction log-likelihood using the approximation in (Wallach et al., 2009) which is now the standard. We compare against the FJS algorithm. Figure 4(upper) summarizes the results for different strategies in generating the pairwise comparisons with $K = 10$ held fixed. The log-likelihood is normalized by the total number of pairwise comparisons tested. As depicted in Fig. 4 (upper), the log-likelihood produced by the proposed algorithm RP is higher, by a large margin, compared to FJS. The predictive accuracy is robust to how the comparison data is constructed. We also consider the normalized log-likelihood as function of K (see Fig. 5). The results validate the superior performance and suggest that $K = 10$ is a reasonable parameter choice.

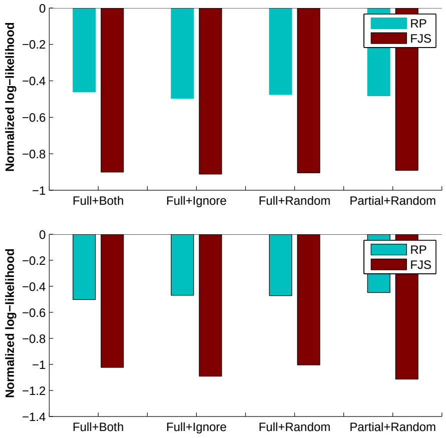

Figure 4: The normalized log-likelihood under different settings for (upper) new comparison prediction and (lower) new user prediction on the truncated Movielens. $K = 10$.

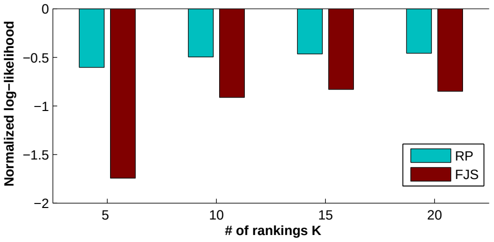

Figure 5: The normalized log-likelihood for Full + Ignore strategy for various K on the truncated Movielens dataset (new comparison prediction).

[3] We show in the appendix that Alg. FJS can be applied to our generative scheme since it only uses the first order statistics, and all the technical conditions are satisfied.

[4] The training/testing split is available at `http://www.cs.toronto.edu/~rsalakhu/BPMF.html`

New user prediction: In this setting, all the ratings of a subset of users are used to generate the training comparisons while the remaining users' comparisons are used for testing. Following (Lu and Boutilier, 2011), we split the first 4000 users (in the original dataset) in the Movielens dataset for training, and the remaining for testing. We use the held-out log-likelihood, i.e., $\Pr(\mathbf{w}_{test}|\widehat{\boldsymbol{\sigma}})$ to measure the performance. The log-likelihoods are again calculated using the standard Gibbs Sampling approximation (Wallach et al., 2009). We compare our algorithm RP with the FJS algorithm. The log-likelihoods are then normalized by the total number of comparisons in the testing phase. We fix the number of rankings at $K = 10$. The results which are summarized in Fig. 4 (lower) agree with the results of the previous task.

6.4 Movielens - Rating prediction via ranking model

The purpose of this experiment is to **illustrate** that our ranking model can capture real-world user behavior through rating predictions, one important task in personal recommendation (Toscher et al., 2009). We first train our ranking model using the training comparisons, and then predict ratings based on comparison prediction. Our objective is to demonstrate results comparable to the state-of-the-art rating-based methods rather than achieving the best possible performance on certain datasets.

We use the same training/testing rating split from (Salakhutdinov and Mnih, 2008a) as used in *new comparison prediction* in Sec. 6.3, and focus only on the $Q = 100$ most rated movies. We first convert the training ratings into training comparisons (for each user, all pairs of movies she rated in the training set are converted into comparisons based on the stars and the ties are ignored) and train a ranking model. The prior is set to be Dirichlet.

To predict stars from comparison prediction, we propose the following method. Consider the problem of predicting $r_{i,m}$, i.e., the rating of user m on movie i. We assume $r_{i,m} = s, s = 1, \ldots, 5$, then compare it against the ratings on movie $\{j_1, \ldots, j_V\}$ she has rated in training. This generates a set of pairwise comparisons $\mathbf{w}_{i,m}(s)$. For example, if user m has rated movies A, B, C with stars $4, 2, 5$ respectively in the training set and we are predicting her rating s of movie D. Then for $s = 3$, $\mathbf{w}_{D,m}(3) = \{(A, D), (D, B), (C, D)\}$ while for $s = 1$, $\mathbf{w}_{D,m}(1) = \{(A, D), (B, D), (C, D)\}$. We then chose s to maximize the likelihood of $\mathbf{w}_{i,m}(s)$,

$$\hat{r}_{i,m} = \arg\max_s p(\mathbf{w}_{i,m}(s)|\mathbf{w}_{train}, \widehat{\boldsymbol{\sigma}}).$$

We evaluate the performance using root-mean-square-error (RMSE). This is a standard metric in collaborative filtering (Toscher et al., 2009). [5] We compared our ranking-based algorithm, RP , against rating based algorithms. We choose to compare two benchmark algorithms, Probability Matrix Factorization (PMF) in (Salakhutdinov and Mnih, 2008b), and Bayesian probability matrix factorization (BPMF) in (Salakhutdinov and Mnih, 2008a) for their robust empirical performance [6]. Both PMF and BPMF are latent factor models. The number of latent factors K has the similar interpretation as in our ranking model. The RMSE for different choices of K are summarized in Table 2.

Table 2: Testing RMSE on the Movielens dataset

K	PMF	BPMF	RP	BPMF-int
10	1.0491	0.8254	0.8840	0.8723
15	0.9127	0.8236	0.8780	0.8734
20	0.9250	0.8213	0.8721	0.8678

Although coming from a different feature space and modeling perspective, our approach has similar RMSE performance as the rating-based PMF and BPMF. Since the ratings predicted by our algorithm are integers from 1 to 5, we also consider restricting the output of BPMF to be integers (denote as BPMF-int). This is achieved by rounding the real-valued prediction of BPMF to the nearest integer from 1 to 5. We observe that our RP algorithm outperforms PMF which is known for over-fitting issues, and matches the performance of BPMF-int. This demonstrates that our approach is in fact suitable for modeling real-world user behavior.

We point out that one can potentially improve these results by designing a better comparison generating strategy, ranking prior, aggregation strategies, etc. This is, however, beyond the scope of this paper.

We note that our proposed algorithm can be naturally parallelized in a distributed database for web scale problems as demonstrated in (Ding et al., 2014). The statistical efficiency of the centralized version can be retained with an insignificant communication cost.

Acknowledgment

This article is based upon work supported by the U.S. AFOSR and the U.S. NSF under award numbers # FA9550-10-1-0458 (subaward # A1795) and # 1218992 respectively. The views and conclusions contained in this article are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the agencies.

[5] Normalized Discounted Cumulative Gain (nDCG) is another standard metric. It requires, however, to predict a total ranking and is inapplicable in our test setting.

[6] The implementation is available at `http://www.cs.toronto.edu/~rsalakhu/BPMF.html`

References

S. Arora, R. Ge, and A. Moitra. Learning topic models – going beyond SVD. In *Proc. of the IEEE 53rd Annual Symposium on Foundations of Computer Science*, New Brunswick, NJ, USA, Oct. 2012.

S. Arora, R. Ge, Y. Halpern, D. Mimno, A. Moitra, D. Sontag, Y. Wu, and M. Zhu. A practical algorithm for topic modeling with provable guarantees. In *Proc. of the 30th International Conference on Machine Learning*, Atlanta, GA, USA, Jun. 2013.

P. Awasthi, A. Blum, O. Sheffet, and A. .Vijayaraghavan. Learning mixtures of ranking models. In *Advances in Neural Information Processing Systems*. Montreal, Canada, Dec. 2014.

H. Azari Soufiani, H. Diao, Z. Lai, and D. C. Parkes. Generalized random utility models with multiple types. In *Advances in Neural Information Processing Systems*, pages 73–81. Lake Tahoe, NV, USA, Dec. 2013.

D. Blei. Probabilistic topic models. *Commun. of the ACM*, 55(4):77–84, 2012.

W. Ding, M. H. Rohban, P. Ishwar, and V. Saligrama. Topic discovery through data dependent and random projections. In *Proc. of the 30th International Conference on Machine Learning*, Atlanta, GA, USA, Jun. 2013.

W. Ding, M. H. Rohban, P. Ishwar, and V. Saligrama. Efficient Distributed Topic Modeling with Provable Guarantees. In *Proc. ot the 17th International Conference on Artificial Intelligence and Statistics*, Reykjavik, Iceland, Apr. 2014.

D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *Advances in Neural Information Processing Systems 16*, pages 1141–1148, Cambridge, MA, 2004. MIT press.

V. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In *Advances in Neural Information Processing Systems*. Vancouver, Canada, Dec. 2009.

D. F. Gleich and L.-H. Lim. Rank aggregation via nuclear norm minimization. In *Proc. of the 17th ACM International Conference on Knowledge Discovery and Data Mining*, pages 60–68, San Diego, CA, USA, 2011.

S. Jagabathula and D. Shah. Inferring rankings under constrained sensing. In *Advances in Neural Information Processing Systems*, pages 753–760. Vancouver, Canada, Dec. 2008.

T. Lu and C. Boutilier. Learning mallows models with pairwise preferences. In *Proc. of the 28th International Conference on Machine Learning*, Bellevue, WA, USA, Jun. 2011.

C. L. Mallows. Non-null ranking models. i. *Biometrika*, pages 114–130, 1957.

S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, pages 2474–2482. Lake Tahoe, NV, USA, Dec. 2012.

S. Oh and D. Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, Montreal, Canada, Dec. 2014.

B. Osting, C. Brune, and S. Osher. Enhanced statistical rankings via targested data collection. In *Proc. of the 30th International Conference on Machine Learning*, pages 489–497, Atlanta, GA, USA, Jun. 2013.

R. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

T. Qin, X. Geng, and T.-Y. Liu. A new probabilistic model for rank aggregation. In *Advances in Neural Information Processing Systems*, pages 1948–1956. Vancouver, Canada, Dec. 2010.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proc. of the 31st International Conference on Machine Learning*, Beijing, China, Jun. 2014.

F. Ricci, L. Rokach, and B. Shapira. *Introduction to recommender systems handbook*. Springer, 2011.

R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proc. of the 25th International Conference on Machine Learning*, pages 880–887, Helsinki, Finland, Jun. 2008a.

R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008b.

A. Toscher, M. Jahrer, and R. M. Bell. The bigchaos solution to the netflix grand prize, 2009.

M. Volkovs and R. Zemel. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15:1135–1176, 2014.

H. M. Wallach, I. Murray, R. Salakhutdinov, and D. Mimno. Evaluation methods for topic models. In *Proc. of the 26th International Conference on Machine Learning*, Montreal, Canada, Jun. 2009.

C. Wang and D. Blei. Collaborative topic modeling for recommending scientific articles. In *Proc. of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 448–456, 2011.